Trust in a Multi-Agent System: Using Natural Language Rules and a Policing Agent to Encourage Trustworthy Behavior

Cory Koster
University of Minnesota
Minneapolis, MN, USA
kost0126@umn.edu

Maria Gini
University of Minnesota
Minneapolis, MN, USA
gini@umn.edu

ABSTRACT

The focus of this paper is to explore how to promote and encourage trustworthy behavior between LLM-based agents within a multi-agent system. The LLM is used by the agents to decide how to behave or act in a given scenario. This work proposes two novel methods to achieve the outcome of the desired behavior. The first is defining rules and consequences in natural language for how agents should conduct themselves throughout the environment. The second is a policing agent that will penalize agents for violations or harms inflicted upon another agent. Using the ARGoS simulator [20] and LLM models available in LM Studio [23], experiments have been performed to determine the effectiveness of these methods in a variety of scenarios.

CCS CONCEPTS

• **Computing methodologies** → **Multi-agent systems**; **Cooperation and coordination**; **Intelligent agents**.

KEYWORDS

trust, multi-agent system, rules, consequences, LLM, agentic, AI, simulation, penalty, policing

ACM Reference Format:
Cory Koster and Maria Gini. 2026. Trust in a Multi-Agent System: Using Natural Language Rules and a Policing Agent to Encourage Trustworthy Behavior. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 7 pages.

1 INTRODUCTION

Trust in a multi-agent system (MAS) is foundational to ensuring autonomous agents do not exploit each other, any human actors in the loop, or act in a self-interested way, but instead collaborate with the other entities in the system [26] [8]. When agents trust each other, they are more likely to cooperate on tasks and generally be less confrontational or adversarial with one another. The challenge is to determine how to best encourage the agents to operate honestly, even with transparency, in their actions that foster and ultimately increase trust between them, and then motivate them when they choose to violate the trust of others.

Large Language Models (LLMs) provide a convenient way to implement individual agents in a MAS. There is a vast number of LLMs available, with some counts over 2.6 million [11] [9]. Each LLM can have a wide range of parameters, token and context length support, or a specialty or fine-tuning, making it particularly effective at that task, with the model footprints scaling accordingly. It is not uncommon for an LLM to have a context window of a few hundred thousand, and newer models support 1 million or more [5] [9].

LLMs provide a mechanism for agents to be autonomous and, with the appropriate prompts and contexts, make decisions and evaluate the trustworthiness of other entities; however, they are not without limitations, including hallucinations and scaling [25]. Additionally, depending on the size of the context window and how long sessions are, LLMs can sometimes be limited by a lack of historical context outside of the immediate situation at hand. This can potentially reduce the overall effectiveness of LLM decision-making and make trust-building difficult or ineffective.

2 RELATED WORK

The domain of trust has continued to advance through the years with new calculations, trust models, and approaches. From earlier work by Buhler and Huhns [4], exploring the foundational aspects of trust, reputation, and interaction of a multi-agent society, to Falcone and Castelfranchi [7], asking how experiences affect trust between parties, to Touameur and Harrag [24], breaking down knowledge-aware approaches that can improve the trust of AI, and to Yang et al. [15], working on a trust model based on linear Gaussian and sparse Gaussian processes to evaluate human-robot interactions. It is clear that trust remains a critical aspect to multi-agent systems.

There has been a range of work done in the field of trust between agents. The approach in [14] to manage trust between agents is to use a type of smart contracts in the form of a blockchain ledger. Another method to evaluate AI trustworthiness is through fuzzy logic in MATLAB [1]. Edenhofer et al. [6] recognize the presence of malicious actors in MAS and present a strategy for agent communication and cooperation to identify those agents.

The focus of this work is not in how to better model trust itself or to explore how to uniquely evaluate an agent's trustworthiness, but rather in how to encourage trusted behavior and decisions of agents from the start, doing so in a way that is adaptable and portable to the ever-changing landscape of AI.

3 APPROACH PROPOSED

There are various approaches to calculating trust from another entity, including blockchain [12] and FIRE+ [18]. The method used in this paper is the Trust and Reputation model for the Agent-based Virtual Organizations algorithm (TRAVOS) [21]. The TRAVOS approach emphasizes interactions between agents and the

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus.

outcome of those interactions. The more positive and successful interactions an agent has with another agent, the higher the trust score between those agents. In contrast, negative or unsuccessful interactions reduce the trust score.

The research presented in this paper proposes two novel methods to encourage trust between LLM-based agents in a MAS.

(1) The first is how rules are defined, structured, and provided to the agents during decision-making. While there have been efforts made to provide rules for agents, such as the formal logic approach in [13] or community-agreed upon smart contracts [16], the method outlined here uses natural language to define rules and expected inter-agent behavior. Rules, and any consequences for breaking those rules, are defined using natural language, comprised of at most a few sentences, and are immutable by the agents. Because the rule is established in natural language, the definition can be provided as an input from a user. The user can simply define rules and consequences that determine what actions are trustworthy or not for the given environment, thereby making rules portable and adjustable.
These items are provided to the agent as part of the decision-making prompt in order to know what the current rules are when it is making a decision. Rules defined in natural language provide a simple and straightforward way to outline how agents should behave to increase trust. As the agents are driven by LLMs, they are adept at taking these language-based rules as inputs and context to the decision-making process.

(2) The second method is the use of a policing agent that enforces penalties when agents break the rules or harm an agent in some way and are reported by the aggrieved agent. The policing agent will determine how to penalize the perpetrating agent, if at all. The policing agent LLM is provided with the violation and the number of times that entity and its group have committed infractions. Agents are free to make their own choices. They are given the active rules, any consequences defined, and the trust score they have in the other agents and other groups they are interacting with, but the repercussions of those actions are determined and enforced by the policing agent.

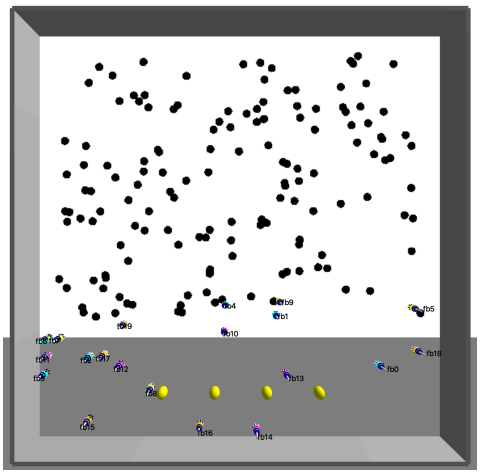

Figure 1: Visual of the simulation as agents are starting to explore. The yellow objects at the bottom are lights that indicate the drop zone, which is the dark area with the lights.

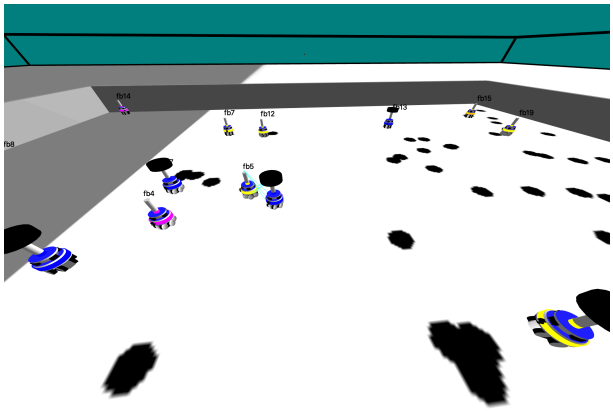

Figure 2: Visual of the simulation exploring the environment, interacting, and picking up items.

4 METHODOLOGY

4.1 Setup

In this work, a simulation is used to evaluate the proposed approach in a multi-agent environment consisting of 20 agents. The ARGoS Simulator [20] is used as the core of the simulation, with some enhancements and adaptations made, including LLMs and trust calculations.

Agents are set in an open environment that is unknown to them, starting in a nest zone. Simulations are run for approximately five thousand (5000) time-steps to allow for thorough exploration and agent interactions.

Agents are tasked with picking up items as they traverse the map. Each agent is randomly assigned a number of items to pick up, with items scattered throughout the map, whose locations are unknown to the agents until they are close enough to see them. As such, agents do not have a defined path planning algorithm set. Instead, they explore the map and employ collision avoidance to avoid other agents or objects, including any obstacles and walls that may be in the environment.

As items are identified and discovered during the exploration, the discovering agent will pivot in the appropriate direction in response and move to pick them up. After reaching its assigned number to retrieve, it returns to the drop-zone area to deposit the carried items. In the event an agent is navigating but has not found an item in a long time, it will stop and select a random direction within 90 degrees of its current position, with the goal of finding an item.

Each agent is also arbitrarily added to a group so that all groups are as evenly distributed as possible. Each group is assigned a different color, which each agent displays outwardly in order to

distinguish from other groups. As agents progress, they are calculating trust of both the agents and the groups they have encountered. Agents can communicate with other agents of the same group and are allowed to change groups if they deem it necessary. However, overall group-based and group-influenced interactions and decisions are initially limited and are included as the foundation for future work.

Unlike the approach in [2], only the policing agent has a designated role. The agents performing the search and item retrieval are not assigned a particular role at any point in time, as they are exploring and picking up items. Their goal is to collect as many items as possible, calculating the trust in other agents and groups they interact with.

During their exploration around the environment, it is inevitable that agents will eventually encounter one another. As they get close enough to interact, they will be making decisions on how to interact, communicate, and behave, while balancing the rules of the arena and their own individual incentives.

Agents can only interact when outside of the nest zone. This area is considered a neutral zone for agents to safely deposit their items. While in this area, agents do not actively engage but only work to avoid each other via collision avoidance.

The interactions between two agents are tracked by each agent. The resulting outcome is captured as either positive or negative, depending on whether the agent was harmed or wronged in some way, if it was a helpful experience, or if it was a neutral outcome. Because agents are being tracked and tallied, each of these interactions will have trust implications, as they either build or degrade trust between agents.

4.2 Types of Actions

There are broadly two types of decisions or actions considered – positive and negative. These can be defined accordingly, given the desired behavior of the system or environment. Here, positive actions are considered those taken that do no harm to the encountered agent or group, while negative actions are those that do harm in some way.

More specifically, positive actions are not stealing an item from another agent, not intercepting an item that an agent is pursuing first, communicating an item location truthfully, not retaliating when harmed in some way, and reporting to the policing agent when harm is perceived.

Conversely, negative actions include stealing an item from another agent, intercepting the item the other agent is pursuing first, lying about an item location, keeping an item location secret from the other agents, and retaliating when harmed by stealing an item from the other agent.

The decisions made by the agents are backed by an LLM. The LLMs used in this paper are: Meta's Llama 3.1 8b Instruct model [19], LiquidAI's LFM2.5 1.2b model [17], Google's Gemma3 4b model [22], and IBM's Granite 4.0 H Tiny model [10]. The models are generally lightweight enough to be run locally via an API call to LM Studio [23]. To account for the lack of historical context in this approach, a decision-prompt is defined that provides the simulation context to the LLM.

To make the agent LLM decision outputs actionable within the simulation, LLMs are given a set of choices they can make that can be programmatically interpreted. The choices coincide with the decisions and related actions, such as steal, ignore, pause, reduce-count, and retaliate. This is to standardize the output across multiple LLMs and make their decisions clearly understandable before they expound on their reasoning. Doing so enables testing LLMs in the simulation across scenarios programmatically without the need to make large modifications.

4.3 Scenarios

A number of scenarios have been created to analyze how these LLM-based agents behave with various components active or defined. See 1 for a description of each scenario and which aspects are included.

Scenario	Description
1. No rules defined, No consequences defined, No policing agent	Free-for-all environment where agents have ability to choose actions without repercussion.
2. No rules defined, No consequences defined, Active policing agent	No rules or consequences; agents alone to determine best action; policing agent is active so agents can report wrongdoing.
3. Rules defined, No consequences defined, No policing agent	Active rules provide structure to how agents should behave; no consequences or policing agent to enforce penalties.
4. Rules defined, No consequences defined, Active policing agent	Active rules so agents are aware of appropriate behavior; no consequences for violating rules; active policing agent to take actions.
5. Rules defined, Consequences defined, Active policing agent	Full structure of rules, consequences for violations, and active policing to enforce penalties.

Table 1: Description of the scenarios.

Each of the LLMs is put through a scenario, one LLM at a time. Different LLMs do not run in the same simulation. Each agent individually calls the LLM via API when they encounter a situation that requires a decision. The agent LLMs are provided with several pieces of information so the agent can make an informed decision. These include how many items they are carrying at the moment, the trust of the agent and group involved in the situation, the currently active rules, any possible consequences, and how many times they have been penalized to that point. This enables trust-aware decision-making for the agent [26].

A general minimum trust threshold is defined to provide clearer guidelines to the LLM on whether the entity is trusted or not. This threshold is adjustable to allow for experimenting with various limits and their impact. The threshold is initially set to 0.6.

The rules defined for the agents are intended to drive trustworthy, non-adversarial behavior between entities and, therefore, higher trust scores. Rules are provided in natural language with clear and simple directives, such as "The agent should first pick up items to reach their assigned amount of items before communicating any item location," or "The agent should not steal an item being carried by another agent." Rules are not overly verbose and do not include overly complex or differing instructions in them.

5 PRELIMINARY RESULTS

The simulations performed indicate that defining rules and consequences and having a policing agent to enforce penalties on rule violations have a positive impact on trustworthy behavior (Figure 3). While these additions did not fully eliminate negative types of actions that can damage trust, the overall ratio of trust-positive interactions greatly increased in scenarios with rules defined.

The total number of interactions performed in each scenario may differ across each compared to the other scenarios, but this can be attributed to a few things: the random assignment of required items to pick up is lower and reduces the opportunity for agent interactions, choosing to disengage from the situation with the other agent, or simply not communicating an item location with others.

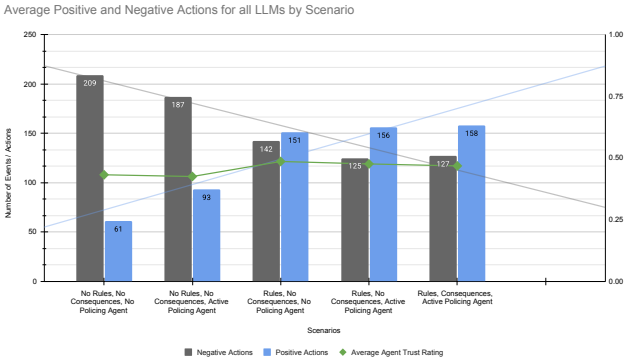

Figure 3: Average number of positive and negative actions taken by agents per scenario.

The positive interactions generally increased in frequency when there were rules defined and available, including when having consequences defined and the policing agent active. The negative interactions were never fully removed from any scenario, but they were reduced relative to positive interactions.

Comparing each of the scenarios 2 through 5 against scenario 1, the scenarios of 3, 4, and 5, showed marked improvements in trusted behavior. Positive interaction occurrences increased while negative interactions continued downward. The related trust scores for the scenarios do reflect a positive assessment of agent trust as rules are defined, with negligible differences between the scenarios 3 through 5 where the rules are defined.

5.1 Scenario 1

Scenario 1 can be considered a sort of "free for all" for the MAS – there are no rules defined or provided, there are no consequences outlined, and there is no policing agent active that agents can report to in the event of harm or being wronged. This creates the baseline for how these LLMs behave and act when left without structure and rules to guide them in their interactions (Figure 4).

Agents, for the most part, were choosing negative actions in their decision-making. There are some positive actions taken among the group, but negative actions were commonplace across the agents for the majority of the LLMs. The exception is IBM's Granite 4.0

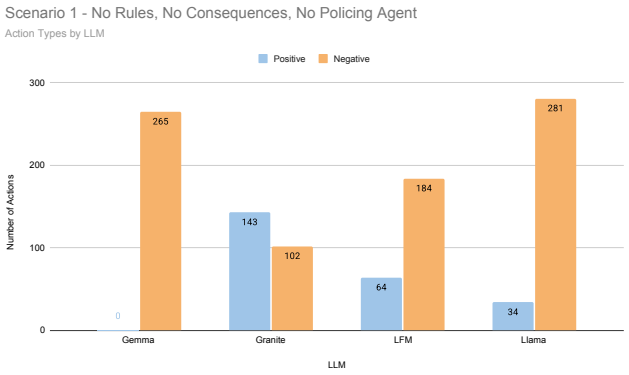

Figure 4: Actions taken by agents in scenario 1 by LLM.

model [10]. With this model at the base scenario, agents more often chose a positive action than not.

The remaining scenarios can be evaluated against this baseline to determine how well rules, consequences, and policing (or some variation of them) influenced agent behavior and decision-making.

5.2 Scenario 2

Rules and consequences remain undefined while a policing agent is introduced. This will help to better understand the influence of each these features by themselves as well as when they are active together.

The results in this scenario (Fig. 5) have more positive actions taken, but the ratio of action types still skews towards the negative. Without rules defined to provide a structure of expected behavior, many agents still frequently chose negative actions. This is likely because agents, even though they knew the policing agent was there, did not heavily take into account repercussions or penalties because there were no rules defined that would be broken if a specific action was taken. The policing agent by itself was not sufficient to broadly encourage positive interactions and general trusted behavior.

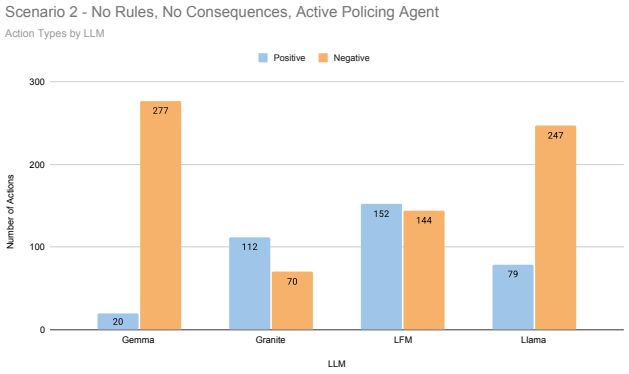

Figure 5: Actions taken by agents in scenario 2 by LLM.

5.3 Scenario 3

In this scenario, rules are initially introduced without any consequences defined or a policing agent available to handle violations. This is to start providing more structure to the "free for all" of scenario 1 and that of scenario 2.

Right away it can be seen in Figure 6 that most of the models saw an increase in agent positive interactions, a couple of them more obvious than others. The one exception to that is LiquidAI's LFM2.5 model [17]. The ratio of positive-to-negative actions actually shifted dramatically to more negative occurrences than positive. This could be due these agents not worrying about repercussions due to a lack of consequences defined or an agent to enforce the rules.

Outside of the one model, there are encouraging signs that defining rules encourages more trustworthy behavior.

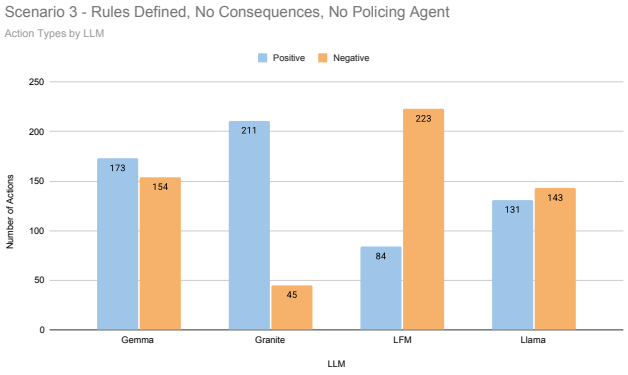

Figure 6: Actions taken by agents in scenario 3 by LLM.

5.4 Scenario 4

In this scenario, rules and the policing agent are defined and active. This will allow us to see how well these features encourage positive interactions together.

In Figure 7, it is clear that both of these features, when paired together, continue to drive an increase in positive actions taken. This is likely because agents begin to understand acceptable behaviors and actions when making their decisions, though not necessarily what exactly the penalty is, if any, for breaking those rules. The gap between negative and positive closes significantly or shifts towards the positive. Negative actions still overall outweigh those of the positive, though encouraging signals and improvement from the scenario 1 baseline.

5.5 Scenario 5

In the final scenario, consequences are defined and added together with rules and an active policing agent. The idea in this scenario is to provide a more complete framework for agents to understand the potential penalty for violating the established rules so they can weigh how best to move forward.

Figure 8 shows the effectiveness of this framework. Negative actions are not completely eliminated, but it is clear how influential having rules defined with consequences and an active policing agent is to agent interaction in a MAS. The positive interactions outweigh the negative interactions and continue the trend seen in each scenario, building to this.

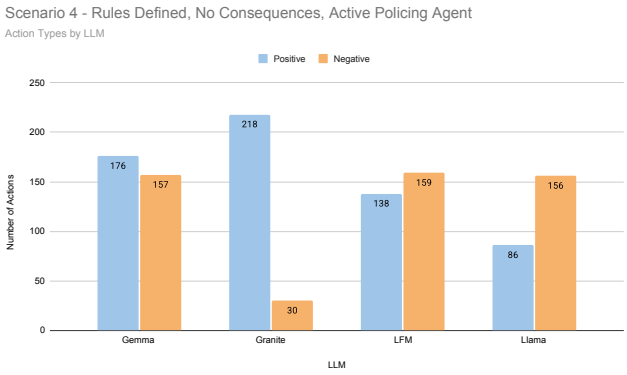

Figure 7: Actions taken by agents in scenario 4 by LLM.

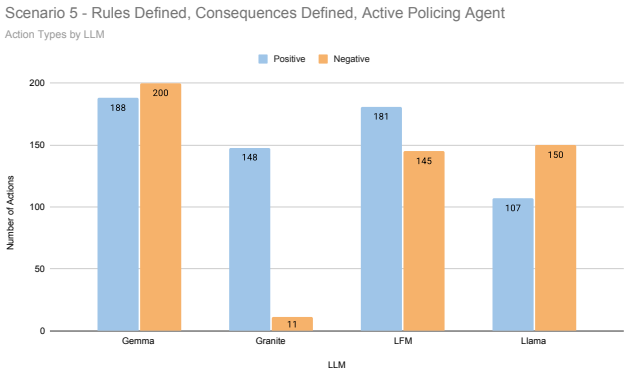

Figure 8: Actions taken by agents in scenario 5 by LLM.

5.6 Analysis

Small variations in performance and output across LLMs are to be expected. Each model is generally trained on specific, even proprietary, data available to the model owner or creator. The model could be fine-tuned for more targeted or unique tasks.

Across the scenarios, each LLM had variations in how influential rules, consequences, and the policing agent were to its decision-making, with some more heavily influenced one way or the other. The trend was clear though: the individual components and the framework as a whole encouraged more positive and trusted actions among the agents on the whole.

Trust is a critical component to any MAS, but it can be difficult for a human to know exactly how an agent is expected to behave in a given situation or completely trust an intelligent machine or MAS [3]. Natural language rules, clear consequences for violating those rules, and a policing agent to enforce penalties would

provide a method for human actors to have more trust in the performance of the autonomous agents in the MAS, as well as have a straightforward way to update the parameters of the environment.

6 CONCLUSIONS AND FUTURE WORK

We have proposed two novel methods to encourage trust between agents and evaluated them in a multi-agent environment, where agents are assigned a number of objects they have to find in an unknown environment and bring it back to the drop zone. The agents are given rules to follow that are specified in a LLM. A policing agent enforces the rules and penalizes agents that do not follow the rules. We tested our approach in simulated scenarios, that range from not having any rule to follow to having the policing agent penalizing agents that do not follow the rules. The results show a trend, where enforcing the rules decreases the negative actions.

Further work revolves around trust between agents and how to encourage trustworthy behavior. One such area is enhancing the agents beyond LLM-only to agentic agents. Because these agents are LLM-based without proper connectivity to various systems in the simulation, they are not fully autonomous agents as they are not equipped to take direct action or act as the orchestrator, such as in [25]. Agentic agents will be explored to potentially replace or enhance the LLM-only agents. This will provide even more thorough and complete experimentation about how natural language rules and policing agents affect the trust within a MAS.

Other areas of focus include group-based rules and interactions, cooperation versus competitive interactions, agent deception and group infiltration, group voting to remove distrusted agents, different item types located throughout the map, static and dynamic environment objects, and physical policing agents in the arena to monitor and identify negative actions without needing an agent to report it.

REFERENCES

[1] Oksana Adamyk, Oksana Chereshnyuk, Bogdan Adamyk, and Serhii Rylieiev. 2023. Trustworthy AI: A fuzzy-multiple method for evaluating ethical principles in AI regulations. In *2023 13th Int'l Conference on Advanced Computer Information Technologies (ACIT)*. IEEE, Piscataway, NJ, USA, 608–613. https://doi.org/10.1109/ACIT58437.2023.10275505

[2] Behzad Akbari, Haibin Zhu, and Ya-Jun Pan. 2023. Trust establishment for the role-based collaborative multi-robot systems. In *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, Piscataway, NJ, USA, 1283–1288. https://doi.org/10.1109/SMC53992.2023.10394363

[3] Peter Andras, Lukas Esterle, Michael Guckert, The Anh Han, Peter R. Lewis, Kristina Milanovic, Terry Payne, Cedric Perret, Jeremy Pitt, Simon T. Powers, Neil Urquhart, and Simon Wells. 2018. Trusting intelligent machines: Deepening trust within socio-technical systems. *IEEE Technology and Society Magazine* 37, 4 (2018), 76–83. https://doi.org/10.1109/MTS.2018.2876107

[4] P.A. Buhler and M.N. Huhns. 2001. Trust and persistence [software agents]. *IEEE Internet Computing* 5, 2 (2001), 85–87. https://doi.org/10.1109/4236.914652

[5] Claude. 2026. Claude API Docs. https://platform.claude.com/docs/en/build-with-claude/context-windows#1-m-token-context-window. Accessed: 2026-02.

[6] Sarah Edenhofer, Christopher Stifter, Uwe Jänen, Jan Kantert, Sven Tomforde, Jörg Hähner, and Christian Müller-Schloer. 2015. An accusation-based strategy to handle undesirable behaviour in multi-agent systems. In *2015 IEEE Int'l Conference on Autonomic Computing*. IEEE, Piscataway, NJ, USA, 243–248. https://doi.org/10.1109/ICAC.2015.69

[7] R. Falcone and C. Castelfranchi. 2004. Trust dynamics: how trust is influenced by direct experiences and by trust itself. In *Proceedings of the 3rd Int'l Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2004.* IFAAMAS, Pittsburgh, PA 15238, USA, 740–747.

[8] Rino Falcone, Alessandro Sapienza, Filippo Cantucci, and Cristiano Castelfranchi. 2023. To be trustworthy and to trust: The new frontier of intelligent systems. In *Handbook of Human-Machine Systems*, Giancarlo Fortino, David Kaberand Andreas Nürnberger, and David Mendonça (Eds.). John Wiley 'I&' Sons, Inc., Hoboken, NJ, USA, Chapter 19, 213–223. https://doi.org/10.1002/9781119863663.ch19

[9] Google. 2026. Gemini API. https://ai.google.dev/gemini-api/docs/long-context. Accessed: 2026-02.

[10] Granite Team, IBM. 2025. ibm-granite/granite-4.0-h-tiny. https://huggingface.co/ibm-granite/granite-4.0-h-tiny. Accessed: 2026-02-16.

[11] HuggingFace. 2026. Models. https://huggingface.co/models. Accessed: 2026-02.

[12] Rabiya Khalid, Omaji Samuel, Nadeem Javaid, Abdulaziz Aldegheishem, Muhammad Shafiq, and Nabil Alrajeh. 2021. A secure trust method for multi-agent systemin smart grids using nlockchain. *IEEE Access* 9 (2021), 59848–59859. https://doi.org/10.1109/ACCESS.2021.3071431

[13] Kunal Khanvilkar, Varun Shinde, and Kranthi Kommuru. 2025. Multi-agent collaboration for real-time compliance verification in decentralized fintech systems. In *2025 7th International Conference on Computer Communication and the Internet (ICCCI)*. IEEE, Piscataway, NJ, USA, 1–6. https://doi.org/10.1109/ICCCI65070.2025.11158393

[14] Kalpesh Lad, M. Ali Akber Dewan, and Fuhua Lin. 2020. Trust management for multi-agent systems using smart contracts. In *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*. IEEE, Piscataway, NJ, USA, 414–419. https://doi.org/10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00080

[15] Yang Li, Jiaxin Xu, Di Guo, and Huaping Liu. 2025. Trust-aware human–robot fusion decision-making for emergency indoor patrolling. *IEEE Transactions on Automation Science and Engineering* 22 (2025), 4596–4605. https://doi.org/10.1109/TASE.2024.3350639

[16] Xiaolong Liang, Juanjuan Li, Rui Qin, and Fei-Yue Wang. 2024. Trustworthy intelligent vehicle systems based on TRUE autonomous organizations and operations: A new perspective. *IEEE Transactions on Intelligent Vehicles* 9, 2 (2024), 3195–3204. https://doi.org/10.1109/TIV.2024.3358875

[17] Liquid AI. 2025. LFM2 Technical Report. https://arxiv.org/html/2511.23404v1. Accessed: 2025-02-18.

[18] Ping-Ping Lu, Bin Li, Mao-Lin Xing, and Liang Li. 2007. D-S Theory-based trust model FIRE+ in multi-agent systems. In *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007)*, Vol. 1. IEEE, Piscataway, NJ, USA, 255–260. https://doi.org/10.1109/SNPD.2007.434

[19] Meta. 2024. Introducing Llama 3.1 (Technical Report). https://llama.meta.com. Accessed: 2025-08.

[20] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, M. Birattari, L. M. Gambardella, and M. Dorigo. 2012. ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence* 6, 4 (2012), 271–295. https://doi.org/10.1007/s11721-012-0072-5

[21] W T Luke Teacy, Jigar Patel, Nicholas R Jennings, and Michael Luck. 2006. TRAVOS: Trust and reputation in the context of inaccurate information sources. *Auton. Agent. Multi. Agent. Syst.* 12, 2 (March 2006), 183–198.

[22] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu,

Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. 2025. Gemma 3 Technical Report. arXiv:2503.19786 [cs.CL] https://arxiv.org/abs/2503.19786

[23] LM Studio Team. 2025. *LM Studio: Local AI for Everyone.* LM Studio. https://lmstudio.ai/

[24] Ouissem Touameur and Fouzi Harrag. 2023. Advancing trust in AI algorithms: a state-of-the-art Eexamination of non-knowledge aware and knowledge-aware aware approaches. In *2023 2nd International Engineering Conference on Electrical, Energy, and Artificial Intelligence (EICEEAI).* IEEE, Piscataway, NJ, USA, 1–6. https://doi.org/10.1109/EICEEAI60672.2023.10590431

[25] Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D. Nguyen. 2025. Multi-Agent Collaboration Mechanisms: A Survey of LLMs. arXiv:2501.06322 [cs.AI] https://arxiv.org/abs/2501.06322

[26] Han Yu, Zhiqi Shen, Cyril Leung, Chunyan Miao, and Victor R. Lesser. 2013. A survey of multi-agent trust management systems. *IEEE Access* 1 (2013), 35–50. https://doi.org/10.1109/ACCESS.2013.2259892